Preprint of: H. Berghel, **D. Berleant**, T. Foy*, and M. McGuire*, “Cyberbrowsing: information customization on the Web,” *Journal of the American Society for Information Science (JASIS)* **50** (10) (May, 1999), pp. 505-511.

Cyberbrowsing: Information Customization on the Web

Hal Berghel*, Daniel Berleant*, Thomas Foy+, Marcus McGuire#

* University of Arkansas, +Metro Information Services, #Raytheon E-Systems

ABSTRACT

The ability to discriminate and distinguish individual documents among the ever increasing volumes of information available through the digital networks is becoming more and more difficult. With Websites being added to the 100 million installed base by tens of thousands per month, information overload is inevitable [1]. There are two basic environments for dealing with information overload: filtering [14] information before it reaches the end-user, and customizing [3-6] the information after it arrives. Filtering remains primarily a server side activity since filtering at the client-side would necessitate unnecessary downloads. Information customization, on the other hand, is basically a client-side activity designed to pick up where information filtering leaves off.

In this article, we describe our vision of information customization and, along the way, chronicle the development of our proof-of-concept prototype, Cyberbrowser.

THE INFORMATION CUSTOMIZATION CONJECTURE

The long-term effectiveness of the technique of information customization described here and elsewhere [4-7] is related to the "information customization conjecture," which holds that information filtering technology will never be able to keep up with the volume of possibly relevant information - something of a "Boyles Law" for cyberspace. Put another way, this conjecture claims that information filtering and related automated techniques may never be able to reduce the dimension of available data to levels which are within the bounds of the typical end-user's personal "bandwidth." Our experience with the Internet thus far is a confirming instance of this conjecture.

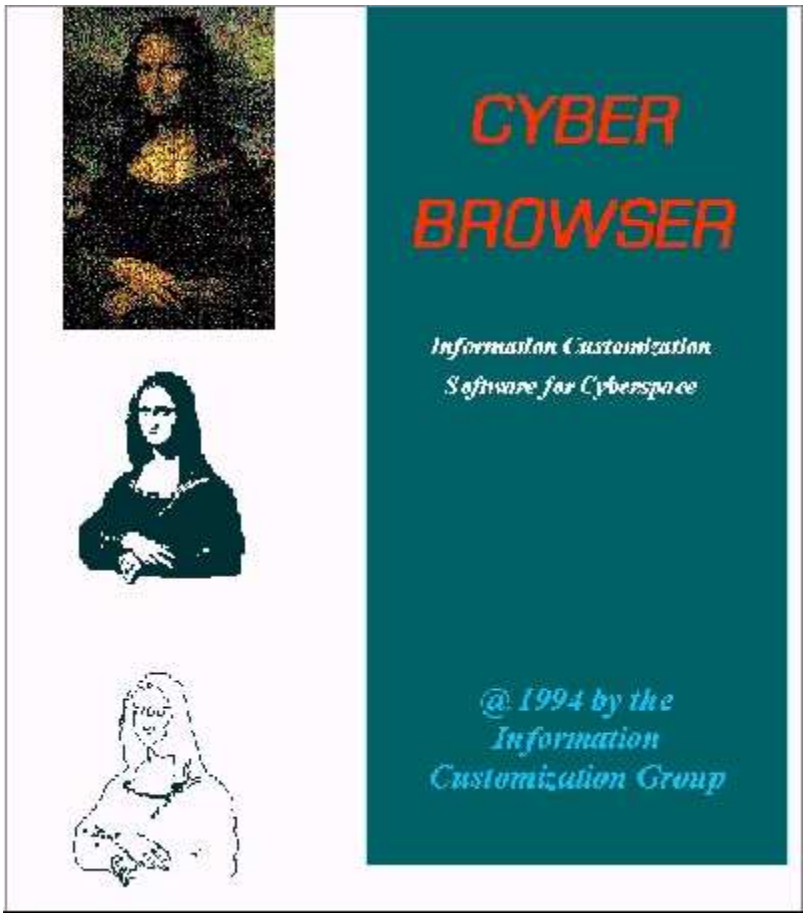
Information overload on digital networks may be thought of as a river network with the hydrologic cycle reversed. Huge volumes of water travel from the oceans through the mouth to the tributaries, streams and headwaters. The operative part of the analogy is the constraint on the velocity of flow as the main channels move waters upstream into ever-smaller tributaries. In our analogy, the streams will never be able to handle all of the reversed flow. Information filters can be thought of as digital levees in our hypothetical model. They can be effective for moderate volumes, but, because of the local topology, they are of minimal value against widespread flooding.

Our belief is that in most cases there will always be the risk that the digital networks will send more information "downstream" than the end-user can consume. Existing information retrieval and filtering technology allows a user to modify the view of document space to immediate needs, but a further step is needed to customize the documents themselves, transforming them into a form compliant to the user's requirements. We have referred to such transformations as Information Customization [op cit] and have discussed elsewhere how it might be applied to text 4-6 and graphics [7] . In this paper, we'll discuss how the techniques of information customization may be built into a network client.

THE INFORMATION CUSTOMIZATION METAPHOR

The goal of information customization is streamlined access to, and absorption of, information by users. While information customization can be done by humans for other humans (information brokerage), the future of information customization lies in automation.

In our view, information customization is the latest element in the evolutionary chain of digital information handling technologies. Tools, techniques and operational metaphors have been developed for information storage, transfer, distribution, acquisition and agency and brokerage through the mid-1990's. Information customization assumes that all of these techniques, taken together, can still produce information overload for the end-user, and that the optimal solution to the remaining problem is highly-interactive, client-side, network-enabled software. This view is conveyed in the splash page of one of our information customization prototypes (see below [external file = "figure0.jpg"]). We attempt to reduce the "depth" of the data, in this case a graphic, in order to ease the uptake of the information without significant loss of meaning.



As we write, network oriented document acquisition is redefining itself. Early strategies were modeled after techniques used to distribute digital information on magnetic media. By the late 1970's network deflectors were serving in the role that magnetic media duplication had served years earlier, and early bulletin boards originally satisfied the same needs as archived collections of programs and data. File Transfer Protocol (FTP), together with compatible document indexing and downloading tools like Gopher and Wide Area Information System (WAIS), marked a transition in the redefinition of network acquisition tools. Although these tools were useful, they were fundamentally impaired for general-purpose network information exchange because the "browsing" was limited to system-specific information (e.g., long file names, directory names, path structures). In retrospect, we now see that FTP indexing was a clumsy technique based on the same metaphor as the physical distribution of magnetic information --- fetches based not on the content of the document, but on its label, title, file name or location.

More than any other single technological event, Telnet showed us the way to achieving our information retrieval objectives. Telnet gave the digital networks a virtuality they had not previously enjoyed, and enclosed networked computers in a unifying cyberspace. With Telnet, networked computers became extensions of our desktop, and it wasn't long before Tim Berners-Lee and his colleagues at CERN came up with specifications for Internet protocols that would provide platform-independent support for distributed multimedia on the Internet. The World

Wide Web was born, and with it came the concept of a navigator-browser. By early 1995, Merit NIC reported that the Web had become the leading packet hauler on the Internet.

While the multimedia, hyper-linked structure of the Web allows users to search for information they need more efficiently than before, the Web alone is still sub-optimal with respect to information acquisition and distribution. One of the reasons for this is that the search and filtering processes were added as afterthoughts, and not built into the original Web design. As an illustration, the content descriptor tag didn't become part of the HTML standard until 1995! This tag is actually the "business part" of the HTML header and drives is used by many Web spiders, wanderers and worms which are the heart of modern search engines. In addition, specifications for information filtering and automated information agency, and information customization for that matter, are independent of the specifications for either HTTP or HTML. At this writing the advanced information acquisition/distribution tools on the one hand, and Web utilities on the other, are developing more-or-less independently of one another thereby diminishing their potential.

Further, search engines (at both the meta and object levels) are inherently ill-equipped to deal with high-recall, high-precision information distribution and acquisition. They work most efficiently when information is indexed, graded and categorized prior to posting, which is rarely done. Even the simplistic META CONTENT= tag seems to be ignored in most documents. Since the Web didn't grow out of the philosophy of pre-processing before posting, there is a definite practical limit to the performance that one may expect of search engines in the foreseeable future, no matter how finely tuned. After-the-fact natural language understanding utilities remain elusive.

In general, effective document location and identification technology is becoming an increasingly indispensable link to the world of information for the modern professional. But as powerful as these tools are becoming, they are intrinsically limited in their support of the information consumer once information has arrived. Thus transfer of knowledge from the computer to the user is more of a bottleneck than ever before. Even client-side systems such as Bellcore's SuperBook Browser, and Digital Equipment's Lectern system, remain oriented to the information provider. It should be emphasized at this point that information customization attempts to deal with this problem by orienting itself to the information consumer.

Information customization complements existing information acquisition, distribution and agent/broker tools and increases their effectiveness. It has five basic characteristics: (1) it is best performed on the client side, (2) it is specifically designed to maximize information uptake, rather than filter or retrieve, (3) it "personalizes" documents by such techniques as extraction, (4) it is normally done interactively, through a "document dialog," and (5) the capability of *non-prescriptive*, non-linear document traversal is provided by the software. Condition (2) sets information customization apart from traditional information filtering and retrieval, while (4) would set it apart from information agency, and (5) would distinguish it from traditional non-linear document traversal systems (e.g. hypertext).

In operation, information customization programs transform an information entity --- such as a document or a set of documents --- into a form that suits the needs of a particular user at a particular moment. This central intuition has considerable currency. For example, information

customization is similar to Englebart's view control [9] and Nelson's concept of transclusion [10] , as well as having strong connections to data mining [8] and knowledge discovery [11].

One can think of the ideal information customizer as taking as input a triple containing a purpose, a cognitive context, and information to customize, and producing as output that processed form of the information which is best attuned to the user. The purpose may be fleeting. The cognitive context changes continually. Only the information to customize is likely to have some constancy, and when that information is a mailing list archive (ala Hypermail) or an institutional knowledge base (cf. Lotus Notes) even that is no longer a given.

Since the point of information customization is to help people absorb the right information more quickly, an obvious strategy is to provide them with the information they need, withholding the information they don't need, and to provide them with that information in a user-friendly way that promotes its absorption. Furthermore, since information customization becomes more important as useful information artifacts become more accessible, information customization becomes most important in an age in which large quantities of relevant documents or other information artifacts are electronically available to the user. Thus the information filters, Web-based search tools, and digital libraries of today and tomorrow make information customization tools increasingly indispensable.

THE INFORMATION CUSTOMIZATION ARCHITECTURE

In general, information customization would involve an interactive process whereby users would interactively and in real time control the means by which documents were reduced in size, or transformed into a more useful form, and displayed. Figure 2 illustrates this process in our current proof-of-concept prototype, Cyberbrowser, which behaves as either a stand-alone application or a browser-compliant, spawnable peruser (i.e., helper app). We are currently working on plug-in and Java prototypes which incorporate the same features.

We have identified a number of features and design specifications that appear to contribute positively to the quality of an information customization system. Continued progress in architectural guidelines for information customization systems requires more research and

empirical observation to better understand these and other issues, including tradeoffs and other interactions. The following are desirable architectural features of information customization tools:

- *Document Dialog*. Interactive information customization helps the user guide the information traversal or transformation process so that it results in the most purposeful custom presentation. The uniqueness of the customizing moment dictates that the interface should allow the user to customize with respect to immediate, perhaps transient, interests and needs.
- *Interface Transparency*. Users of information customizers should concentrate on information absorption, not on manipulating the interface. Therefore it is important that the interface be unobtrusive [13], to avoid interfering with the goal of maximizing the efficiency of information transfer to the user. Such a transparent interface should be

intuitive, and provide a small number of powerful options to avoid distracting the user from the primary pursuit of information uptake.

- *Input Format Independence*. The user should not need to be concerned with the data format (plaintext, HTML, WP, etc.) of a document or other information artifact to be customized. Making format considerations invisible to the user is implied by the basic requirement that information customization provide information in a way that is suited to the user. Our current interests emphasize compatibility with existing Web tools conforming to the de facto HTTP and HTML standards. This is both because the nuances of compliance with mature desktop protocols are research-indifferent but resource-intensive, and because of the intrinsic importance of the Web.
- *Multiway Lookahead*. What a user most needs to see next is likely to be related to what the user is seeing currently. However there are likely to be numerous related items in the document or other information artifact. Multiway lookahead means computing several related items the user might want to see next and displaying them simultaneously. The user can then simply go on to read one or more of the precomputed items without the mental overhead of thinking about requesting what to see next. We have a proof-of-concept prototype, MultiBrowser, which is based on multiway lookahead.
- *Non-Insularity*. Information customizing services will be most useful when used with other systems that provide information. When mature, information customization tools will be menu items of everyday word processors, desktop publishing software, Web navigator/browsers, and so forth. They will complement the existing client-server base, including a wide variety of client-server browsers, locators, emailers, and transfer programs (cf. [2]). The client server base will provide information distribution back ends to customizing software.
- *Nonlinearity*. By nonlinearity we mean that the order of presentation of information from a document or other information artifact is not determined by its physical or digital layout. A paradigm nonlinear viewing environment is hypertext, which is an essential element of the Web's HTML language definition and document layout strategy. We see nonlinearity as factoring into information customization tools in two ways. First, in the nonlinear traversal of a document during interactive perusal that extends the way that hypertext viewers work today. Second, in the creation of nonlinear extracts or browse traces of the document, such as for later hardcopy perusal. Passages of the document should often be grouped based on custom content considerations although the layout of the original document may define different groupings. For example, the customization of an article might involve selecting only those passages most related to the reader's current project. Or a document extract could be dynamically created in a size customized to the user's time constraints. Enabling users to get right to those (perhaps dispersed) parts of a document that are most relevant to their current needs promotes both the authors' goals in influencing their readers, and the readers' goals in finding the information most relevant to their needs.
- *Nonprescriptiveness*. By nonprescriptiveness we mean the ability to transform or traverse an information artifact in ways which were not prescribed by the information provider. In contrast, hypertext is typically a prescriptive environment: the anchors and links in a hypertext document are typically prescribed by the author and hard-coded into the document. While this allows for non-linear traversal, it is prescriptive. Nonprescriptiveness means that the document may be traversed or processed in useful

ways that were unforeseen and perhaps even unintended by its creator. This makes it possible for an information artifact to be more flexibly customized. Nonprescriptiveness is akin to Englebart's "Every object intrinsically addressable" concept [9].

- *Real Time Performance*. The whole point of information customization is to speed up the transfer of useful information to a user's mind, so making a user wait for the system to compute obviously detracts from the performance of an information customizer. Real time performance is all the more important in highly interactive settings where information to be presented is transient and must be recomputed frequently.

Improved understanding of these and perhaps other architectural principles, their interactions, and their conditions of application, relate not only to our own immediate research goals but also to important related work such as visual data navigation, Web resource locators, database mining, and others.

CYBERBROWSER

Our earliest work with information customization began in the late 1980's with concurrent investigations into digitally "simplifying" both images [7] and in the early 1990's, with text [4-6]. Our goal was to find ways in which we could reduce the volume and dimension of data which were accessible to the consumer via the digital networks. Part of this work led to a prototype information customization desktop utility called "Keychain," which allowed end-users to transform the presentation of a document according to intersecting keyword chains detected in the document by pre-processing. The successor prototype, "Schemer," added hypertext capabilities and expanded the range of control that the user would have over the presentation while at the same time adding on-the-fly pre-processing.

Throughout the development of Keychain and Schemer, we viewed information customization as a desktop-centric utility for accelerated content discovery. By 1993, when the World Wide Web was becoming popular, we had changed our view of information customization to include network-centricity as well. The current prototype, Cyberbrowser, extended our initial design philosophy to include:

- accommodating unprocessed text files, especially including TXT, ASCII and HTML
- adding a simple, intuitive, Windows-look-alike interface
- implementation of additional document extraction operations
- compatibility with mainstream Web browsers (Netscape, Internet Explorer, Mosaic)
- use of a histogram, instead of text, to display keyword frequencies
- the option to view either the customized extract of the original document by itself, or the complete original document with the customized portion highlighted
- creation of a text analysis program which can analyze raw text files offline and store for later use
- implementation of additional logical operations including the "extract kernel" and "extract meta-kernel" operations. The kernel of a document is the collection of sentences which contain the greatest number of keywords. The meta-kernel sentences of a document are the kernel sentences of the five most frequently occurring keywords. The concept of document kernel is actually a place-holder for cluster of document analysis

algorithms which would be available in a commercial-grade product., and are not intended to be complete or exhaustive.

- A text analysis program was also created which may be run when needed to analyze raw text files and store the information needed for the document customization application in an appropriate form in the interest of speed. The processing and the format of the pre-processing will be optimized for use with CyberBrowser.

The Figures below shows a typical control view in CyberBrowser. Figure 1 (Figure 2) depicts a typical, keyword-based (sentence-based) document extraction.

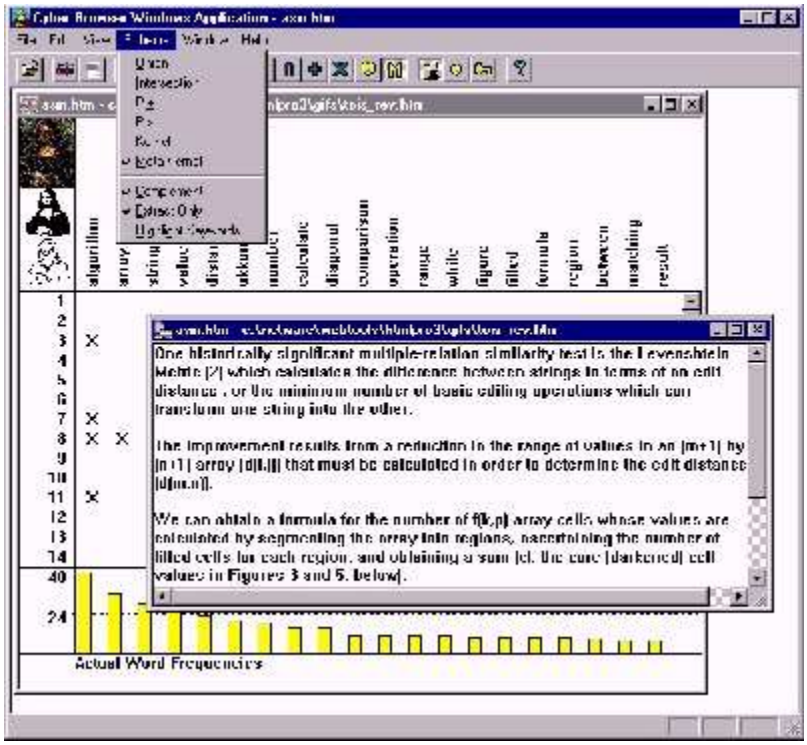


Figure 1: Keyword-oriented document extraction

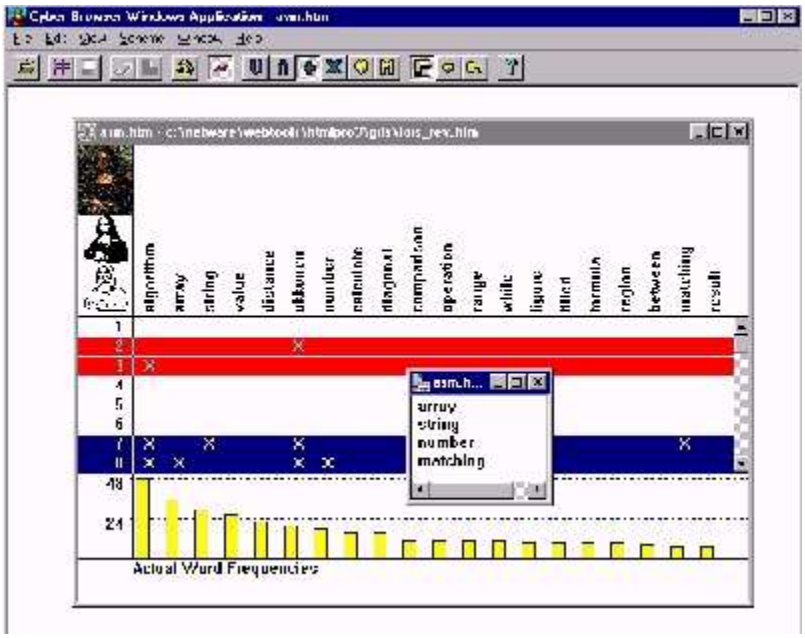


Figure 2: Sentence-oriented keyword extraction

Note that in both Figures, the presence of a keyword (top row) in a sentence number (left column) is indicated by an "x". The bar chart at bottom plots the absolute frequency of the keywords. (We have experimented with relative frequency measures based upon comparisons with standardized corpora (e.g., the Brown Corpus) as well as various weighted measures, but deprecated these functions in the latest version of the prototype). This operation was invoked by clicking the histogram button (see below).

The red and blue bars which identify keywords (vertically) or individual sentences (horizontally) show that a manual document extraction is being performed on those keywords or sentences which have (blue) and do not have (red), have the respective keywords. Since the complement button has been clicked, the complemented keyword selections are included in the calculations performed when creating the document extract. Were this feature not enabled, the complemented selections are treated as if the complemented items had not been selected at all. Document extracts by sentences produce keyword sets; extracts by keywords produce sets of sentences.

Cyberbrowsing functionality may be categorized in terms of presentation schemes, the underlying text algebra, and document extraction techniques. We define them in Table 1, below, by reference to the items on the button bar.

TABLE 1: Cyberbrowsing Functionality by Category

Presentation Schemes


View Control - keyword mode. Display the n most common keywords in the document and plot them against the sentences (by number) in which they appear.


View Control - extract mode. Display the document extract called for by the query


View Reset. Erase all keyword selections and restart document analysis in current window


View Histogram. View the absolute frequency of keyword distribution in current document. (Earlier releases of Cyberbrowser offered relative and weighted frequency measures as well).


Adjust Weighting. Change weights assigned to keywords which were found in HTML <TITLE> tags, HTML <META> tags, or document <BODY>

Text Algebra


Enable the underlined components of the text algebraic functions below.


Produce a document extract which contain at least one of the selected keywords, and none of the complemented keywords


Produce a document abstract which contains all selected keywords, and no complemented keywords


Projection of a set of keywords which occur in selected sentences, but do not occur in any complemented sentences


Projection of a set of keywords which occur in all selected sentences, but do not occur in any complemented sentences

Extraction Modes


Extract kernel sentences - e.g., the union of the sentences which contain the greatest number of different keywords, and none of the complemented keywords


Extract meta-kernel sentences - e.g., the kernel sentences for the k most frequently occurring keywords


Context toggle - if enabled, only extract will be displayed; else, full text is displayed with kernel sentences highlighted


Set granularity - modifies the number of sentences on each side of a target sentence which

will be included in the abstract - more means added context.


key highlighting mode - if enabled, keywords will be highlighted in the display of document, and all hyperlinks will be active..

CONCLUSION

Information customization is becoming increasingly important as modern information access methods make overwhelming quantities of information electronically available to the individual user.

In this paper, we have outlined some architectural considerations of what we believe will be successful information customization programs. We also provide a functional overview of our own information customization prototype, Cyberbrowser, which is the latest of three generations of our client-side, information customization programs. Cyberbrowser is designed to supplement existing Windows desktop programs as well as integrate fully with Web navigation/browsing clients.

Our research program has a three-pronged plan of attack. (1) Develop novel and distinct information customization systems need to facilitate continued progress in the theory and practice of the information customization field; (2) integrate such systems with digital network (especially, the World Wide Web) and digital library technology; and (3) Advance the concept and theory of information customization as a general information systems paradigm.

At this writing, we are extending Cyberbrowser in several ways. First, we have developed a prototype, HyperBrowser, which dynamically adds links to plaintext or HTML documents so that sentences become linked to other sentences within the document through keyword chains. This prototype contrasts with typical hypertext conversion systems (e.g. Hypermail) in that the number and density of HyperBrowser links will normally be higher, to maximize the nonprescriptiveness of the system. HyperBrowser viewing is accomplished through standard Web browsers, and all of the original document links persist under transformation.

Second, another prototype, MultiBrowser, allows the user to concentrate on the material they are interested in with minimal distraction from the interface. By making available to users what they will want to read next, without their having to explicitly click for it or otherwise request it, the user interface will demand significantly less attention and manipulation. Unfortunately, what a user will want to read next is obviously not fully predictable. Therefore, we are working on systems which present several likely alternatives simultaneously. This approach is termed multiway lookahead. Discussions of the HyperBrowser and MultiBrowser information customization extensions will be deferred to a future forum.

We expect that information customization, in due course, will take its place along complementary information technologies and play a useful role in dealing with information overload.

REFERENCES

[1] Berghel, H., "Cyberspace 2000: Dealing with Information Overload", Communications of the ACM, 40:2 (1997), pp. 19-24

[2] Berghel, H.: "The Client Side of the Web", Communications of the ACM, 39:1 (1996), pp. 30-40.

[3] Berghel, H. and D. Berleant, "The Challenge of Customizing Cybermedia", *The Journal of Knowledge Engineering & Technology*, 7:2, Summer/Fall 1994, pp. 33-34.D.

[4]Berleant, D. and H. Berghel, "Customizing information: Part1, Getting what we need, when we need it", *Computer*, September 1994, pp. 96-98.

[5] D. Berleant and H. Berghel, "Customizing Information: Part 2, How Successful are we so far?", *Computer*, October 1994, pp. 76-78.

[6] Berleant, D. and H. Berghel, "Electronic Information Management: A Perspective", Proceedings of the 1993 Arkansas Computer Conference (1993), pp. 50-57.

[7] Berghel, H., D. Roach, and Y. Cheng, "Expert Systems and Image Analysis", *Expert Systems: Planning, Implementation, Integration*, Summer 1991, pp. 45-52.

[8]Fayyad, U., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.

[9] Englebart, D. C., Toward Augmenting the Human Intellect and Boosting our Collective IQ, Communications of the ACM 38:8 (Aug. 1995) p. 30 ff.

[10] Nelson, T. H., The Heart of Connection: Hypermedia Unified by Transclusion, Communications of the ACM, 38:8 (Aug. 1995) p. 31 ff.

[11] Piatetsky-Shapiro, G., and W. Frawley, eds., Knowledge Discovery in Databases, MIT Press, 1991.

[12] Ribarsky, W., E. Ayers, J. Eble, and S. Mukherjea, Glyphmaker: Creating Customized Visualizations of Complex Data, Computer 27:7 (July 1994) 57-64.

[13] Wieser, M., The World is Not a Desktop, Interactions 1:1 (Jan. 1994) 7-8.

[14] Special section on Information Filtering, Communications of the ACM 35:12 (Dec. 1992) 26-81.